What is claimed is:

1. A pipeline arrangement for a network traffic scheduler comprising:

A hierarchical structure;

A plurality of SRAM and DRAM memory devices external to the scheduler;

Control blocks of scheduling elements stored in said memory devices with at least some of the memory devices storing more than one type of control block; wherein

a) SRAM memory is used if the content of a control block is Read-Modify-Write at packet enqueue and at dequeue;

b) SRAM and DRAM memory are used if the control block content is Read-Modify-Write only, at the packet dequeue; and

c) DRAM memory is used if the control block content is Read only at packet enqueue and dequeue.

2. The arrangement according to claim 1 wherein DRAM memory is preferentially used if the Read-Modify-Write content is only at the packet dequeue.

3. The pipeline arrangement according to claim 1 wherein the control blocks include flow queue control blocks, frame control flow blocks, hierarchy control blocks, target port queue control blocks, hierarchy control blocks and schedule control blocks.

4. The pipeline arrangement according to claim 1 wherein the hierarchical structure comprises a physical port bandwidth that is divided into a plurality of logical links, the bandwidth available to each of the logical links is divided into a plurality of VLANs, and the bandwidth associated with each VLAN is shared by a plurality of individual user flows.

5. The pipeline arrangement according to claim 1 wherein the pipeline arrangement also includes non-hierarchical link sharing whereby physical port bandwidth resources are shared among individual traffic flows.

6. The pipeline arrangement according to claim 1 wherein memory access to enqueue tasks does not conflict with memory access allocated to dequeue tasks.

7. The pipeline arrangement according to claim 4 wherein a first SRAM contains a set of data buses and stores flow queue control blocks that are required for 'read' and 'write' at each flow queue enqueue and dequeue time, and a second SRAM containing two data buses, one dedicated to 'read' and one dedicated to 'write', and stores frame control blocks and VLAN hierarchy control blocks.

8. The pipeline arrangement according to claim 7 wherein a first DRAM stores a network management counter and a second DRAM stores flow queue control blocks and VLAN hierarchy control blocks that require 'read' only at both enqueue time and dequeue time.

9. A method for retrieving and pipelining information for a network traffic scheduler, wherein the information is stored in a plurality of SRAM and DRAM devices comprising storing functional queue control blocks in the SRAM and DRAM devices, wherein control block content that is Read-Modify-Write at both packet enqueue and dequeue time is stored in SRAM devices; control block content that is Read-Modify-Write packet only at dequeue time is stored in either SRAM or DRAM devices; and control block content that is read only either at enqueue or dequeue time is stored in DRAM devices.

10. The method according to claim 9 wherein information is in two modes, hierarchical link sharing and non-hierarchical link sharing, and the structure for hierarchical sharing comprises a physical port bandwidth which is divided into a plurality of logical links, the bandwidth available to each of the logical links is divided into a plurality of VLANs, and the bandwidth associated with each VLAN is shared by a plurality of individual user flows.

11. The method according to claim 10 wherein physical port bandwidth resources for non-hierarchical links are shared among individual traffic flows.

12. The structure for a hardware scheduler pipeline comprising a plurality of control blocks; a plurality of memory devices external to the scheduler in which the control blocks are stored, at least some of the memory devices sharing more than one type of control block; and a hierarchical pipeline arrangement for link resource sharing serving multiple queues.

13. The structure according to claim 12 wherein the hierarchical pipeline arrangement sharing serves the multiple queues within an approximated fixed period of time.

14. The structure according to claim12 wherein the control blocks include, from the group consisting of flow queue control blocks, frame control blocks, calendar control blocks, target port queue control blocks, calendar control blocks and hierarchy control blocks.

15. The structure according to claim 14 wherein the control blocks that are accessed less frequently within a fixed period of time are stored in DRAM memories, and control blocks that are accessed with higher frequency within a fixed period of time are stored in SRAM.

16. The structure according to claim 15 wherein basic flow QCB is stored in SRAM memory and dequeue/enqueue read only flow QCB is stored in DRAM memory.

17. The structure according to claim 13 wherein the hierarchical arrangement comprises a physical port bandwidth that is divided into a plurality of logical links, the bandwidth available to each of the logical links is divided into a plurality of VLANs, and

the bandwidth associated with each VLAN is shared by a plurality of individual user flows.

18. The structure according to claim 17 including time-based calendar arrays for guaranteed bandwidth service and weighted fair queueing calendar arrays for allocation of available bandwidth for competing flows at each port when no service is required by the time-based calendars.

19. The structure according to claim 18 wherein each of the calendar arrays contains three pointers comprising a current position pointer, a current time pointer and a next position pointer.

20. The structure according to claim 18 wherein the time-based calendar provides a scheduling function for flow queues and VLANs.

21. The structure according to claim 18 wherein the weighted fair queueing calendar provides best effort scheduling in flow queues, VLANs and logical links.

22. An article of manufacture comprising a computer usable medium having a computer readable program embodied in said medium, wherein the computer readable program, when executed on a computer, causes the scheduler to store functional queue control blocks in external memory storage devices comprising a mix of SRAM and DRAM devices based on the block content at enqueue and dequeue time, and to share the external devices among the control blocks.

23. The article according to claim 22 wherein the program causes a control block content having Read-Modify-Write at both enqueue and dequeue time to be stored in SRAM; a control block content having Read-Modify-Write at only dequeue time to be

stored in either SRAM or DRAM; and a control block content having 'read' only to be stored in DRAM.

24. The article according to claim 23 wherein the program causes the scheduler to select a flow queue to egress for each duration of a scheduler tick using a time-based calendar or a weighted fair queueing calendar.